**Data Quality Improvement for Medical Concept Normalization**

Krishna Vikas Meduri  
Data Science  
University of North TexasDenton, Texas, USA  
krishnavikasmeduri@my.unt.edu

Rashmi Jakkani  
Data Science  
University of North TexasDenton, Texas, USA  
rashmijakkani@my.unt.edu

Harith Mote  
Data Science  
University of North TexasDenton, Texas, USA  
harithmote@my.unt.edu

Prudhvi Narayan Korupolu  
Data Science  
University of North Texas Denton, Texas, USA  
PrudhviNarayanKorupolu@my.unt.e

*Abstract*—Automatically recognizing medical concepts mentioned in social media texts are used to enhance the people health quality in a community. However, the discrepancy between the language used in social media (such as twitter and Facebook) and the medical concepts have been challenging. Evaluated the approaches using three different datasets, where social media texts are extracted from twitter or Facebook messages. Poor data quality has also direct impact on the machine learning systems. This first, exposed from a dataset that has hidden problem that is used to built a machine learning concepts. Due to the poor quality of the dataset and the defective validation process the results of the experiment showed the great performance. The results also demonstrated that the evaluation of data quality is useful for guiding the quality improvement of machine learning. Therefore, proposed a data quality evaluation framework that includes the quality and their corresponding approaches.

The code and the data for this project are available on GitHub at:

Keywords—Data Quality, Medical concept normalization, Exploratory Data Analysis, Model implementation, Agreement rate, Data Annotation.

# Introduction

The data present in social media can be used to improve the understanding of patience experience in healthcare such as the spread of infectious diseases and side-effects of drugs. However, the social media text and medical concepts text have discrepancy between them and that has become very challenging. Particularly, the frequent use of informal language and abbreviation forms, as well as the social media messages has taken into account by effective information extraction systems.

Deep learning as the most foremost vital breakthrough in machine learning history that has drawn extraordinary consideration from academics as well as business. Be as that it may, missing of high quality preparing information gets to be a major risk to the utilization of deep learning. Approaches, such as crowdsourcing web information or exchanging information from other spaces have been proposed for upgrading preparing information. By the by, this information seems present commotions, such as invalid information and name clamors. Although some carefully outlined profound learning models are vigorous to massive name commotions, the computing rule the show of “garbage in garbage out” is still pertinent to profound learning.

Rajpurkar detailed a profound learning framework called ChexNet that was developed for diagnosing pneumonia illness based on chest X-ray pictures. They claimed that the ChexNet surpassed normal radiologist execution on pneumonia discovery on both sensitivity and specificity. Many radiologist and machine learning experts doubted the result due to the improper dataset. The problem is due to the lacking approval of the measurable methods with genuine information within the fMRI thinks about. Numerous tests have shown that information with destitute quality may contrarily affect the performance of profound learning altogether. Other experiments have illustrated that way better quality of training data might move forward the execution of profound learning. Hence, an efficient assessment of the quality of the dataset is critical for building a high-quality machine learning framework. The evaluation result would offer rules for information enhancement and framework execution enhancement.

One of the purposes of using this is to improve the understanding the impact of data quality on the performance of machine learning systems. Informative quality is a multi-dimensional concept on subjective and quantitative properties of information, and the definition changes beneath diverse settings.

In this article, we characterize data quality as a estimation of data for fitting the reason of building a machine learning framework. We begin with explored an overclaimed execution enhancement of a machine learning framework. The issue was due to the destitute quality of the datasets for building the framework and the risky approval handle. At last, we presented a system for evaluating the information quality to guarantee the quality necessities of datasets for building a high-quality machine learning framework. It was built on profound learning models, which require a huge sum of preparing information. The datasets displayed in this article appear deficiently and there’s a large parcel of cover between the preparing and the test information.

# Literature Review

The plant specimens in worldwide herbaria are a valuable record of the world’s biodiversity [1][2][3]. The data, which are critical to understanding species distribution, phenology, and identity, are recorded on specimen sheets, stored in the word’s herbaria. To make these data more accessible and amenable to analysis and aggregation [4], many herbaria have undertaken the monumental task of digitizing all herbarium specimens by capturing a digital image of the full specimen sheet, and extracting the typewritten or handwritten data from the sheet into structured data formats. The sheer number of natural history collections has introduced a digitization bottleneck [5][6] which has required the natural history collections community to develop innovative approaches to achieve large-scale digitization.

Citizen science workflows have helped accelerate digitization of specimens through workflows which include human transcription and parsing of specimen data [7][8]. Human transcribers can read and interpret text that is inscrutable to OCR processes and human parsers can effectively parse text that is very challenging to natural language processing (NLP) techniques but much of the text and data on herbarium specimens are quite amenable to OCR methods which can produce very accurate text transcriptions.

OCR is often an integral part of digitization of herbarium specimens [9] and can be used as a source for further human parsing or machine parsing or can be used to cluster records for more rapid, manual transcription [10][11]. High quality OCR transcriptions are more amenable to parsing using NLP techniques [12][13][14]. But, while OCR quality has increased with technological advancements, OCR still can produce results which are not amenable to parsing [10]. The iDigBio augmenting OCR (A-OCR) working group was formed in 2011 to investigate tools and techniques to maximize the potential of these technologies [15][16], and to address the challenges of generating quality OCR results which can be parsed by NLP.

Techniques for segmenting specimen images into regions of interest (ROIs) can be used to contain and isolate regions containing text. These image regions can then be processed using OCR (rather than the whole sheet) to improve OCR quality [16][17][18]. Much of the research for segmenting herbarium specimen images has focused on isolating the plant material from other elements on the specimen sheet (such as labels, stamps, scale bars, etc.) for the purpose of automated species identification [19][20][21][22][23]. Further research in computer vision has been applied to analyzing herbarium specimen images to determine leaf and flower traits [24] segmenting color targets, scale bars, and barcodes [25][26][27].

# Methods

OCR and metadata from the research dataset were be analyzed to address the three research questions. Each method outlined below produced metrics which allow us to quantify the quality of the OCR text produced by each OCR platform and allow us to compare the metrics between whole-sheet OCR and targeted OCR of the primary label region to determine which combination of OCR platform and OCR strategy produces the highest quality text. The metrics for each research question are visualized using the Python package Matplotlib [28].

# Data Collection and Cleaning

Existing specimen records with complete metadata represent the “gold standard” of quality records for the evaluated dataset. This existing metadata is used to determine the completeness and quality of data extracted from OCR text generated from specimen images. Specimen records were selected from various herbarium collection datasets that have both specimen records and images available online through the Symbiota web platform (e.g. https://portal.torcherbaria.org/). Records with associated image records were retrieved using a Python script that allows different query parameters to be used to select records across different herbaria, botanists, species, locations, and other desired traits. The specimen records are structured in Darwin Core format and the image records are structured in Audubon Core format which includes the URL of images allowing the full-resolution image for each record to be retrieved. Approximately 1100 specimen records were retrieved at this stage.

1. A typical specimen image showing regions of interest (ROIs) which contain relevant text (green) in the primary label, and text which is irrelevant (red) to the data extraction process.

Specimen image records acquired through the specimen data download process include URLs of the specimen images related to each specimen record. These images were retrieved using a script to consolidate them into local data storage. The images are the basis for all OCR extracted for this research. Additionally, each image were segmented using a technique developed under a parallel research project which has developed a TensorFlow [29] object detection model using AutoML and has achieved an average precision of 99.6% for detecting primary labels in herbarium specimen images. Each image was processed using two different OCR engines: Google Cloud Vision OCR and Tesseract producing text output for the full image as well as text output for the primary label region which was detected and segmented using TensorFlow (Fig. 1). These four OCR text outputs and the Darwin Core metadata for each image were the basis for all further analysis as outlined in the Methods section. Because some of the research questions are meant to determine the quality of raw text as generated by various OCR methods, the primary data was minimally cleaned to a point where it can be effectively tokenized. The research questions which require more extensive data cleaning provide details of the process below.

# Experiment and Data analysis

The research dataset was aggregated as described above in Data Collection and Cleaning and as illustrated in figure 2. The data was analyzed as described in the methods for RQ1, RQ2, and RQ3 and produced quality and accuracy metrics which determine the optimal strategy for assessing and optimizing OCR quality. These metrics and the analysis of each are facilitated through visualizations generated using Matplotlib.

1. Experiment workflow showing data collection, ROI extraction, and OCR extraction to generate inputs for research questions.

## RQ1 - OCR quality

In an effort to answer RQ1, we determined the quality of OCR text generated from whole-sheet analysis versus targeted ROI by evaluating the OCR results using a number of NLP techniques. Additionally, we used different OCR engines (Google Cloud Vision and Tesseract) to determine which combination of tools and techniques produce the highest quality results(Fig. 3). We measured OCR quality based on three primary metrics: OCR confidence, spell checking, and garbage detection.

Almost all times, in the context of the language model, a word must be checked for spelling correctness and, if necessary, corrected. Spell checking is a basic requirement in any text processing as it can help evaluate the text quality. For this task, we used three different packages that allow us to perform spell correction.

spellchecker - As the module name defines, this package allows us to perform a spell check and outlines the applicable spelling for the misspelled word. In simple terms, this package finds the possible errors in the provided text and offers a suggestion for its alternatives that might fit the best in any given context. This package is very straightforward to use which includes importing the package, creating an instance of the spellchecker module, and finally calling the method spell as shown in a sample example below:

from spellchecker import SpellChecker # import the module

spell = SpellChecker(language='en')#create the instance

# find those words that may be misspelled

Check = spell.unknown(['Spelli', 'Check', 'is', 'impurtant'])

for word in Check: # run the loop

# Get the one `most likely` spelling

print("Most likely the correct spelling:")

print(spell.correction(word))# call the methods

# Get a list of `likely` options

print("Other possible options for the incorrect spelling:")

print(spell.candidates(word)) # call the methods

print("\n")

TextBlob - The next python library that we implemented for the spell checking in the text processing operation is TextBlob. This module offers the easiest implementation of a basic spelling corrector to correct input text by a simple measure of calling the inbuilt method correct ().

First Import the module and create a TextBlob then use the correct() method to attempt spelling correction.

from textblob import TextBlob

b = TextBlob("hiey checke the speling!")

print(b.correct())

>>> hey check the spelling!

This library also provides a simple API to access its methods and allows users to dive into other common natural language processing tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more. Additionally, TextBlob offers a feature to find the confidence level of a word by collaborating with the spellcheck function.

blob = TextBlob('hello letslytics check the confidence level')

blob.correct()

print(blob.words[1].spellcheck())

>>>>>> [('letslytics', 0.0)]

As we all know, computers cannot understand the contents of an image simply by visualizing as humans perceive them. They instead require a more concrete and organized process to understand them, However, it is important to find the confidence score as it identifies a threshold that separates good data from bad ones. Therefore, for the metric of gathering confidence, we used pytesseract, which is an optical character recognition (OCR) tool for Python. This tool enables users to recognize and read the text embedded in specimen images. To capture the text values from the image, we got the boxes around the words by using the function “Image\_to\_data” as below:

import cv2

import pytesseract

from pytesseract import Output

img = cv2.imread('sample.jpg')

results = pytesseract.image\_to\_data(img, output\_type=Output.DICT) # detect and localize text using Tesseract and the image\_to\_data

print(results.keys())

1. Frequency distribution of total tokens and total dictionary words found in OCR results of full sheet analysis and primary label analysis using Google Cloud Vision.

Once the output is available, and the words are detected we generated confidence score for each detected word to evaluate the OCR quality as below:

# loop over each of the individual text localizations

for i in range(0, len(results["text"])):

# extract the OCR text itself along with the confidence of the

# text localization

text = results["text"][i]#grab the text from the OCR

conf = int(results["conf"][i]) # extracts the confidence of the text localization (the confidence of the detected text).

Garbage detection - we used various methods and tools to detect “garbage” or nonsensical characters in the OCR results. We explored a number of approaches including rules-based methods [30], Markov models [31], and others.

For the final part of the RQ1 requirement, we used the Python module nostril (Nonsense String Evaluator) module which compares an input list of words and identify if a string contains any gibberish value. This module that we plan to use can infer whether a given word or text string is likely to be nonsense or a meaningful text. The module is easy to use and provides a function named nonsense(); which takes a single text string as an argument and returns a Boolean value as a result. Below is a sample example of its use:

from nostril import nonsense

Sample = ['Can', 'yee', 'find', 'the', ‘garbagege','herete']

for s in Sample:

print('{}: {}'.format(s, 'nonsense' if nonsense(s) else ‘Correct’))

>>>>>>

Can: Correct

Yee: nonsense

find: Correct

the: Correct

garbagege: nonsense

herete: nonsense

The other simple approaches that we can use to remove gibberish text that exhibits no pattern is via the use of nltk package. To do so, we would first create an empty list and loop through all the words in the current list. Thereafter, use words. words() from the corpora to check if it is the real world and append all the "non-junk words'' to the empty list and validate the results. Below is the sample to achieve the results:

from nltk.corpus import words

sample = [Whatski, 'Gatbhst', 'dssdsfgert', 'sdsdsdn',\

'dsdsdsdl', 'hello', 'ddfgdsed', 'lets', 'try' , to , ‘detect’,’ghghghsdys’, ‘gibberish’ ,’text’]

#create a empty list

correct = []

for word in sample:

if word in words.words():

correct.append(x)

print(correct)

>>[‘hello’, ‘lets’, ‘try’. ‘to’, ‘detect’, ‘gibberish’, ’text’]

## RQ2 - Analysis of OCR and Human Parsed Text

In order to answer RQ2, we compared the OCR results to the existing structured metadata records retrieved as part of the full specimen dataset. First, we determined the feasibility of matching all structured metadata tokens with raw OCR text using rules-based matching (spaCy) and fuzzy matching (spaczz - https://github.com/gandersen101/spaczz). Second we used NLTK and spaCy to perform named entity recognition to determine the success rate of extracting key textual components including named entities, dates, geographic coordinates and other data which can be correlated with the structured Darwin Core metadata.

Data Preprocessing - Data preprocessing is extremely important for any machine learning model and most basic to be done in NLP. Without cleaning, the dataset generally contains a list of words and characters that the computer cannot comprehend. Text preprocessing includes tokenization, data cleaning, normalization, stemming and lemmatization.

Tokenization: Tokenization is splitting the text containing phrases, sentences or paragraphs into individual terms or words. The OCR data is tokenized using NLTK (Natural Language Toolkit).

from nltk. tokenize import word\_tokenize

from nltk. tokenize import sent tokenize

text = " "

word\_tokenize(text)

sent\_tokenize(text)

There are other techniques that can be used for tokenization like python split () function, Regular expressions(RegEx), Spacy library, Using Keras, Using Gensim. Data Cleaning is using modules pandas and numpy. It also includes removing punctuations, removing stopwords, extra white spaces, special characters etc.

Stemming and Lemmatization - Stemming and Lemmatization are the Normalization techniques that can be used for the further processing of words, text and documents.

# import these modules

from nltk.stem import PorterStemmer

from nltk.tokenize import word\_tokenize

ps = PorterStemmer()

# choose some words to be stemmed

words = ["program", "programs", "programer", "programing", "programers"]

for w in words:

print(w, " : ", ps.stem(w))

# import these modules

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("rocks :", lemmatizer.lemmatize("rocks"))

print("corpora :", lemmatizer.lemmatize("corpora"))

# a denotes adjective in "pos"

print("better :", lemmatizer.lemmatize("better", pos ="a"))

A large amount of data is being digitized and stored day by day. The stored data is unstructured data and it is difficult to detect important information that is required. This is when Named Entity Recognition (NER) comes into picture. Named entity recognition is a natural language processing technique used to extract named entities from the text. These named entities in text are converted to predefined categories and these predefined categories make data search quicker. This is the whole purpose of NER. Optical Character recognition is a technology used to recognize any kind of text (handwritten or printed) in an image. The text that is being scanned now needs to be organized. To structure the text that we got as an output of OCR we use Named Entity Recognition. NER can be performed using different libraries. We are using spacy, NLTK and spaczz.

Spacy is an open-source library used in python. Some of the features provided by spacy are PoS tagging, tokenization, text classification and named entity recognition. There is a default model that is provided by spacy which provides few named entities by default. If required, other than the default entities arbitrary classes can also be added to spacy by training the NER model. Some of the entities that are used here is person, species name, organization, date, geographic location, cardinal.

Firstly, Spacy library should be installed using the pip command. Next, all the required libraries are imported.

from pprint import pprint

import spacy

from spacy import displacy

nlp = spacy.load('en\_core\_web\_sm')

sentence =” Text “

entities= nlp(sentence)

#to print all the entities with iob tags

pprint([ (X, X.ent\_iob\_, X.ent\_type\_) for X in entities] )

#to print just named entities use this code

print("Named entities in this text are\n")

for ent in entities.ents:

print(ent.text,ent.label\_)

# visualize named entities

displacy.render(entities, style='ent', jupyter=True)

NER using NLTK - NLTK is a popular Python library for any sort of natural language processing and computational methods. Firstly, we install the NLTK library. Once this is done we import the necessary libraries and their submodules. The sentences are then tokenized into words. Then each word is tagged with its respective part of speech (PoS). After this ne\_chunk() is used to recognize all the named entities in the sentence.

pip install nltk

import nltk

from nltk.tokenize import word\_tokenize

from nltk.tag import pos\_tag

sent= ''' '''

words= word\_tokenize(sent)

postags=pos\_tag(words)

ne\_tree = nltk.ne\_chunk(postags, binary=False)

## RQ3 - Topic modeling specimens based on text content

Specimen digitization workflows often include a transcription stage in which specimen images are presented to human participants who transcribe and parse the text in the image. Specimen datasets are often combined into thematic sets in order to target particular expertise or interest (location, species, etc.). To facilitate this thematic grouping, we explored the use of topic modeling to answer RQ3. We used Gensim [32] to extract salient topics from the OCR text and to determine which records are good candidates for grouping into a thematic transcription set. We used PyLDAvis, a Python implementation of LDAvis [33], to visualize and explore the results and compared these results with the visualization produced by BERTopic [34].

We followed the steps below to implement our LSA topic modeling. First, we loaded the data that is in the csv format. After the load was completed, we preprocessed the data using some of the common methods like removing stop words, tokenization, lemmatizing, and stemming process. Next, we created a document-term matrix and dictionary of terms. After corpus creation, we then generated a model using LSA. Another extra step that we took to optimize results by identifying an optimum number of topics. We generate coherence scores to determine an optimum number of topics and based on our coherence score we plot the coherence score value. For the purpose of our visualization, we have used the pyLDAvis (Fig. 4 & 5) technique, an interactive LDA visualization python package.

1. Detail of pyLDAvis interface showing topic clusters.
2. Detail of pyLDAvis interface showing revelant terms and frequency.

# Results and Discussion

The OCR quality metrics generated in RQ1 indicate that the Tesseract OCR engine performs very well and is comparable to the Google Cloud Vision engine (Fig. 6). This impressive performance is in contrast to anecdotal experience of one of the authors and is perhaps an indicator of recent improvements in Tesseract. When evaluating spelling accuracy of the OCR results, it was apparent that many legitimate words were not correctly identified when compared to the NLTK corpus. The TextBlob spell checker was able to correctly identify more correct words than found in the NLTK corpus (Fig. 7) but still did not correctly identify a number of correct words. Analysis of full sheet OCR versus primary label OCR revealed that a significant number of non-word tokens as well as irrelevant words outside of the primary label region of interest (Fig. 3) can be excluded from further analysis using NLP techniques.

In order to focus our efforts on quality OCR, the resulting text generated from Google Cloud Vision analysis of the primary label was used for research questions 2 and 3. For RQ2, the named entity extraction process using Spacy revealed that many entities were mis-identified as organizations, many were not identified (dates, locations, people). Due to the very domain-specific nature of the botanical Latin terms used for scientific names, we did not expect NER to recognize these words. This was in fact the case and virtually none of the scientific names were recognized. Another set of critical entity types were poorly extracted: person names and location names were rarely detected. The failure of our initial efforts to extract entities from the raw text in sufficient quantity prevented us from effectively comparing the results to human-parsed data. We hope to have the opportunity to improve on these techniques as described in Future Work.

The topic modeling results from RQ3 indicate that the OCR text from herbarium specimens can be effectively used to cluster related specimens. The sample data were clustered into 10 topics and visualized using pyLDAvis (Fig. 4). The interface provided by pyLDAvis provides the ability to navigate the different topics and view the term frequency within each (Fig. 5).

1. Number of dictionary words found by Google Cloud Vision and Tesseract showing very similar results.
2. Comparision of dictionary words found using NLTK corpus versus dictionary words found using TextBlob spelling evaluator (confidence = 1.0).

# Conclusion

The ongoing improvement of OCR platforms combined with the continual advancement of NLP tools and technologies suggest a promising future for accelerating the process of extracting useful data from specimens in natural history collections. Our research has demonstrated some techniques which can be used to evaluate and improve the quality of OCR generated from herbarium specimens and tools which can be used to parse and further enhance the extraction of structured data. While the NER processes we explored did not provide promising results, we believe the underlying process can be improved and shown to be useful. Topic modeling and exploratory visualization interfaces like pyLDAvis can also provide effect methods for exploring datasets and clustering data into related topics based on geography, taxonomy, dates, and other salient features. This clustering can be used to use group specimens into thematic sets allowing citizen scientists and subject matter experts to focus their efforts on related data.

# Limitations and future work

We speculate that NER methods failed to correctly identify many entities due to the terse phrasing which is common on the primary labels of botanical specimens. NER models which have not been trained on such a dataset would be expected to not perform well. With the successful extraction of raw text using OCR we believe that future efforts can explore the use of existing human-parsed text records to provide the basis for supervised training of an NER model which would be more successful.

The OCR quality evaluation was limited to comparison to built in NLTK and TextBlob spell-check methods. A domain-specific dictionary would help provide better evaluation of tokens because the NLTK and other dictionaries (or word corpora) do not include many of the valid words encountered in these datasets. Our analysis revealed that 19,166 tokens did not match dictionary words. Of these, 580 tokens occurred more than five times and 40 tokens occurred more than 50 times. Given the high frequency of many tokens which did not match dictionary words, these could be assumed to be valid words or abbreviations used in herbarium specimen labels.

# Author Contributions

Jason Best - team lead, writing for introduction, literature review, data capture, experiment and design analysis plan, general editing, code review, scheduling, and leading team meetings

Sai Prudhvi Jiddigam - writing and code for RQ2, participating in team meetings

Satish Maharjan - writing and code for RQ1, participating in team meetings

Birendra Pandey - writing and code for RQ3, participating in team meetings

##### References

1. Baird R (2010) Leveraging the fullest potential of scientific collections through digitization. Biodiversity Informatics 7: 130–136.
2. Chapman AD (2005) Uses of Primary Species-Occurrence Data, version 1.0. Copenhagen: Global Biodiversity Information Facility. 106 pp. ISBN: 87-92020-01-1. Accessible at http://www2.gbif.org/Uses.pdf [March 22, 2012]
3. Pyke GH, Ehrlich PR (2010) Biological collections and ecological/environmental research: a review, some observations and a look to the future. Biological Reviews 85: 247–266. doi: 10.1111/j.1469-185X.2009.00098.x
4. Scoble, M. J. (2010). Natural history collections digitization: Rationale and value. Biodiversity Informatics, 77–80.
5. Beaman, R., Macklin, J., Donoghue, M. J., & Hanken, J. (2006). Overcoming the Digitization Bottleneck in Natural History Collections: A summary report on a workshop held 7-9 September 2006 at Harvard University.
6. Rabeler RK, Macklin JA (2006) Herbarium networks: towards creating a ‘toolkit’ to advance specimen data capture. Collection Forum 21: 223–231.
7. Ellwood, E. R., Dunckel, B. A., Flemons, P., Guralnick, R., Nelson, G., Newman, G., … Mast, A. R. (2015). Accelerating the digitization of biodiversity research specimens through online public participation. BioScience, 65(4), 383–396. https://doi.org/10.1093/biosci/biv005
8. Ellwood, E. R., Kimberly, P., Guralnick, R., Flemons, P., Love, K., Ellis, S., … Mast, A. R. (2018). Worldwide Engagement for Digitizing Biocollections (WeDigBio): The Biocollections Community’s Citizen-Science Space on the Calendar. BioScience, 68(2), 112–124. https://doi.org/10.1093/biosci/bix143
9. Nelson, G., Sweeney, P., Wallace, L. E., Rabeler, R. K., Allard, D., Brown, H., … Mast, A. R. (2015). Digitization Workflows for Flat Sheets and Packets of Plants, Algae, and Fungi. Applications in Plant Sciences, 3(9), 1500065. <https://doi.org/10.3732/apps.1500065>
10. Drinkwater, R. E., Cubey, R. W. N., & Haston, E. M. (2014). The use of Optical Character Recognition (OCR) in the digitisation of herbarium specimen labels. PhytoKeys, 30(38), 15–30. https://doi.org/10.3897/phytokeys.38.7168
11. Haston, E. M., Cubey, R., Pullan, M., Atkins, H., & Harris, D. J. (2012). Developing integrated workflows for the digitisation of herbarium specimens using a modular and scalable approach. ZooKeys, 102(209), 93–102. https://doi.org/10.3897/zookeys.209.3121
12. Wei, Q., & Heidorn, P. B. (2008). Interactive Machine Learning ( IML ) Markup of OCR Generated Text by Exploiting Domain Knowledge : A Biodiversity Case Study. In iConference 2008 (pp. 1–5). Retrieved from <http://hdl.handle.net/2142/15134>
13. Heidorn, P. B., & Zhang, Q. (2013). Label Annotation through Biodiversity Enhanced Learning. IConference 2013 Proceedings. Retrieved from http://hdl.handle.net/2142/42056
14. Barber, A., Lafferty, D., & Landrum, L. R. (2013). The SALIX method: A semi-automated workflow for herbarium specimen digitization. Taxon, 62(3), 581–590. https://doi.org/10.12705/623.16
15. Paul, D. L., & Heidorn, P. B. (2013). Augmenting Optical Character Recognition ( OCR ) for Improved Digitization : Strategies to Access Scientific Data in Natural History Collections. In iConference 2013 (pp. 514–518). https://doi.org/10.9776/13266
16. Anglin, R., Best, J. H., Figueiredo, R., Gilbert, E., Gnanasambandam, N., Gottschalk, S., … Watson, K. (2013). Improving the Character of Optical Character Recognition ( OCR ): iDigBio Augmenting OCR Working Group Seeks Collaborators and Strategies to Improve OCR Output and Parsing of OCR Output for Faster , More Efficient , Cheaper Natural History Collections Spe. In iConference 2013, Fort Worth, TX, USA (pp. 957–964). <https://doi.org/10.9776/13493>
17. Kirchhoff, A., Bügel, U., Santamaria, E., Reimeier, F., Röpert, D., Tebbje, A., … Berendsohn, W. (2018). Toward a service-based workflow for automated information extraction from herbarium, (October). https://doi.org/10.1093/database/bay103
18. Owen, D., Livermore, L., Groom, Q., Hardisty, A., Leegwater, T., van Walsum, M., … Spasić, I. (2020). Towards a scientific workflow featuring Natural Language Processing for the digitisation of natural history collections. Research Ideas and Outcomes, 6(August). https://doi.org/10.3897/rio.6.e55789.suppl1
19. Cope, J. S., Corney, D., Clark, J. Y., Remagnino, P., & Wilkin, P. (2012). Plant species identification using digital morphometrics: A review. Expert Systems with Applications, 39(8), 7562–7573. https://doi.org/10.1016/j.eswa.2012.01.073
20. Little, D. P., Tulig, M., Tan, K. C., Liu, Y., Belongie, S., Kaeser-Chen, C., … Ambrose, B. A. (2020). An algorithm competition for automatic species identification from herbarium specimens. Applications in Plant Sciences, 8(6), 1–5. https://doi.org/10.1002/aps3.11365
21. Pryer, K. M., Tomasi, C., Wang, X., Meineke, E. K., & Windham, M. D. (2020). Using computer vision on herbarium specimen images to discriminate among closely related horsetails (Equisetum). Applications in Plant Sciences, 8(6), 1–8. https://doi.org/10.1002/aps3.11372
22. Schuettpelz, E., Frandsen, P., Dikow, R., Brown, A., Orli, S., Peters, M., … Dorr, L. (2017). Applications of deep convolutional neural networks to digitized natural history collections. Biodiversity Data Journal, 5, e21139. https://doi.org/10.3897/BDJ.5.e21139
23. Unger, J., Merhof, D., & Renner, S. (2016). Computer vision applied to herbarium specimens of German trees : testing the future utility of the millions of herbarium specimen images for automated identification. BMC Evolutionary Biology, 1–7. https://doi.org/10.1186/s12862-016-0827-5
24. Pearson, K. D., Nelson, G., Aronson, M. F. J., Bonnet, P., Brenskelle, L., Davis, C. C., … Soltis, P. S. (2020). Machine learning using digitized herbarium specimens to advance phenological research. BioScience, 70(7), 610–620. https://doi.org/10.1093/biosci/biaa044
25. Ledesma, D. A., Powell, C. A., Shaw, J., & Qin, H. (2020). Enabling automated herbarium sheet image post-processing using neural network models for color reference chart detection. Applications in Plant Sciences, 8(3), 1–7. <https://doi.org/10.1002/aps3.11331>
26. Steinke, K. H., Gehrke, M., Dzido, R. (2011). Object Recognition in Herbarium Specimens. In Perner, P., (Ed.), Advances in Data Mining - Poster and Industry Proceedings. 11th Industrial Conference, ICDM 2011, New York, USA. (pp.1-16). IBaI Publishing
27. Triki, A., Bouaziz, B., Mahdi, W., & Gaikwad, J. (2020). Objects Detection from Digitized Herbarium Specimen based on Improved YOLO V3. VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 4(Visigrapp), 523–529. https://doi.org/10.5220/0009170005230529
28. Hunter, J. D. ,Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
29. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In 12 USENIX Symposium on Operating Systems Design and Implementation OSDI 16 (pp. 265–283).
30. Taghva, K., Nartker, T., Condit, A., & Borsack, J. (2001). Automatic Removal of “Garbage Strings” in OCR Text : An Implementation. The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, 3–8.
31. Lui, M., & Baldwin, T. (2011). Cross-domain Feature Selection for Language Identification. Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011), (4), 553–561.
32. Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 45–50.
33. Sievert, C., & Shirley, K. (2015). LDAvis: A method for visualizing and interpreting topics, 63–70. https://doi.org/10.3115/v1/w14-3110
34. Grootendorst, M. (2020). BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. Zenodo. https://doi.org/10.5281/zenodo.4430182